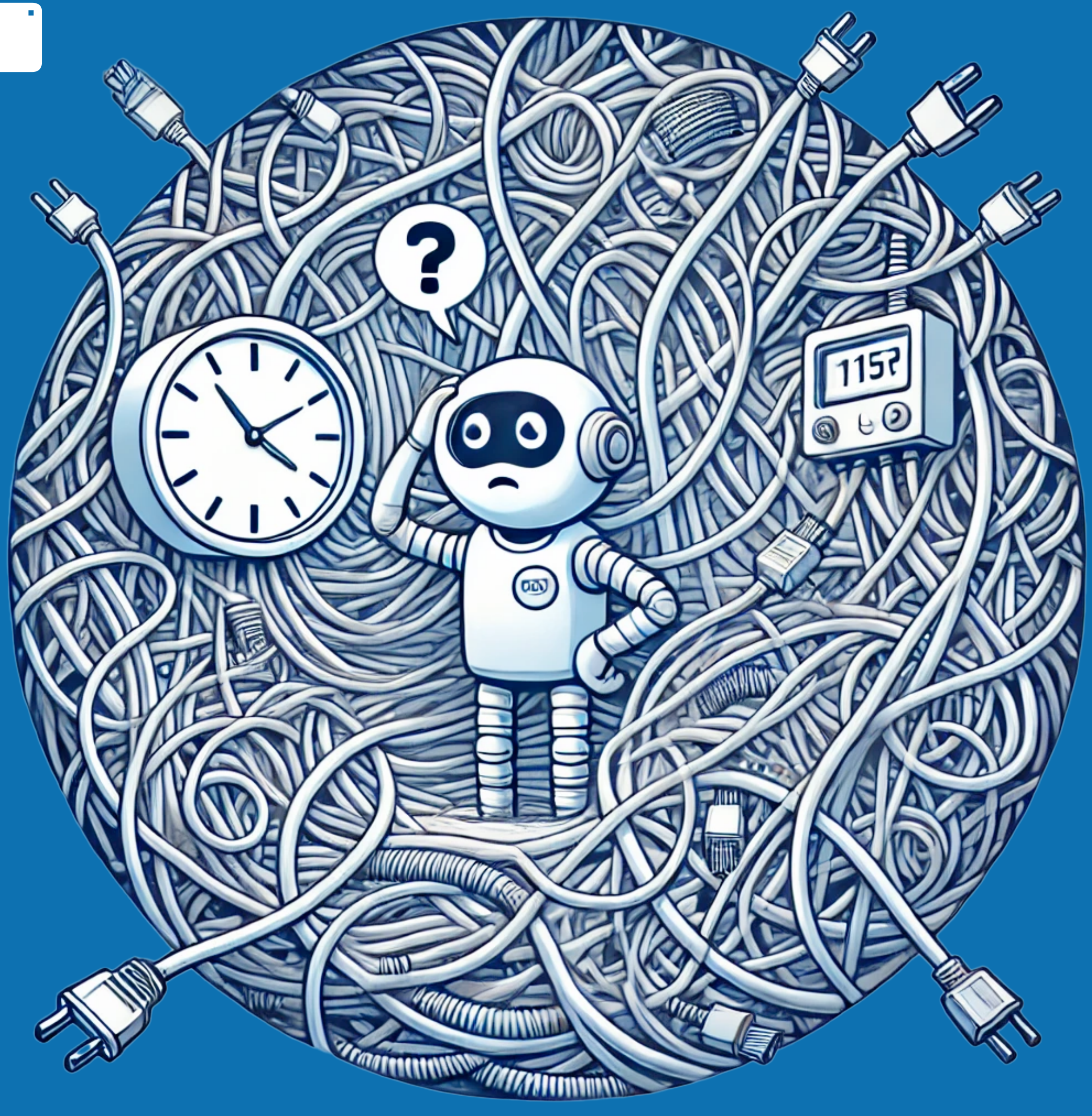
Armin @mitsuhiko Ronacher

THE COMPLEXITY GENIE

lessons learned from making things complex and not setting the right constraints early



Who am I?

I build software — particularly Open Source software — and teams. You might know me as creator of Flask or other Open Source projects.

- Armin @mitsuhiko Ronacher
- mitsuhiko.at

- Bluesky: bsky.app/profile/mitsuhiko.at
- Twitter: x.com/mitsuhiko
- GitHub: github.com/mitsuhiko

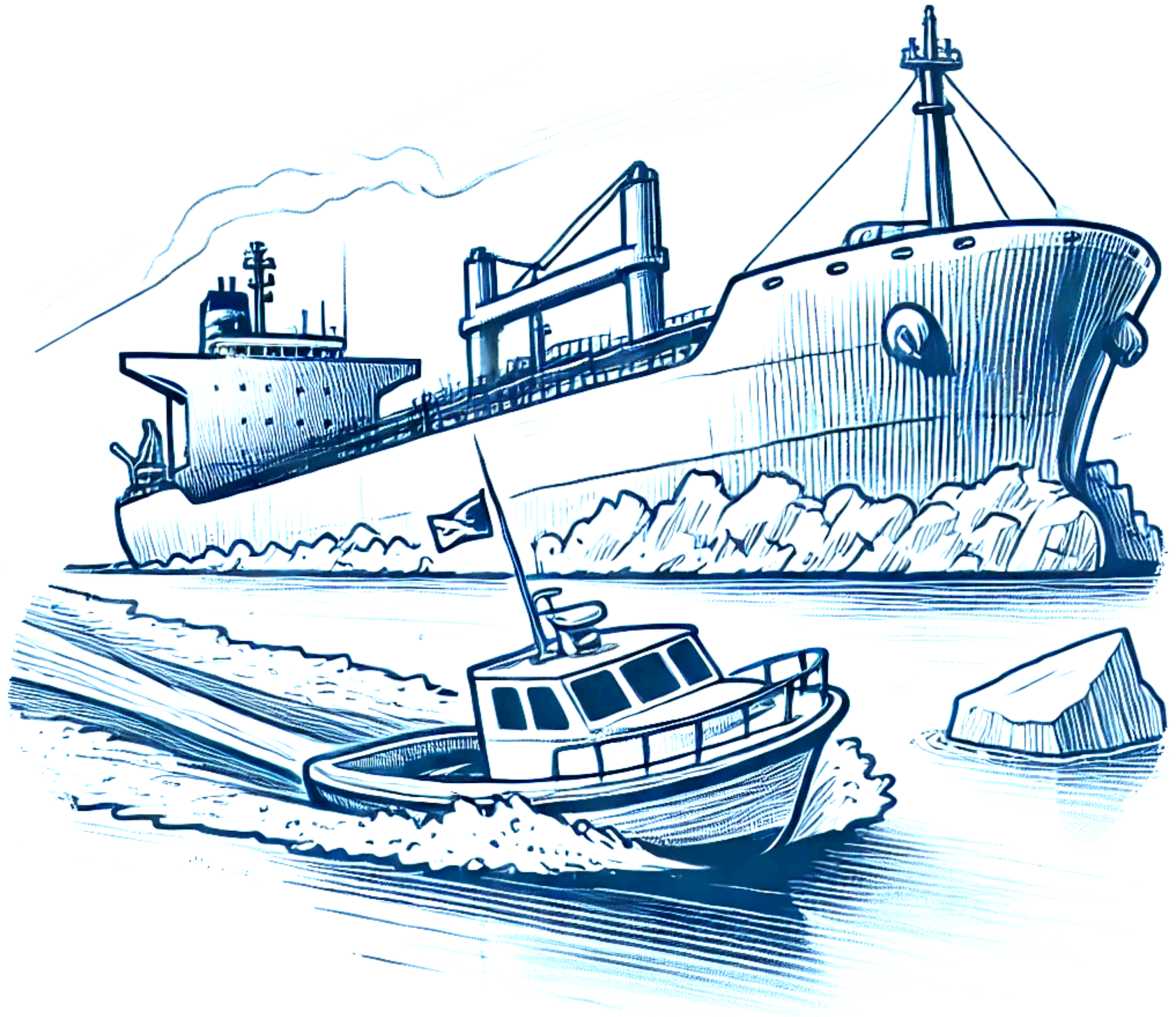
TECHNOLOGY DOESN'T MATTER
BUSINESS OUTCOMES DO

COMPLEXITY IS FELT AS A TAX

IT MAKES EVERY CHANGE MORE EXPENSIVE

NOT EVERY BUSINESS NEEDS AGILITY

- It's okay to be a large, immovable ship
- But then you need to have the force to break the ice
- If you are small, you need the agility
- You need to pick one
- How much are you willing to pay for your complexity?



IF YOU WANT TO FIGHT IT:

TAME COMPLEXITY EARLY ON

WHAT IS COMPLEXITY?

- Something is more complex than necessary or appropriate
- It starts from lack of constraints
- Excessive complexity results in low developer morale and satisfaction

“I could have done this in 15 minutes on my side project but now I'm waiting 3 hours for CI to pass”

IT STARTS WITH LACK OF CONSTRAINTS

CONSTRAINTS ARE GOOD FOR YOU

- Arbitrary constraints are a good start
  - They stop being believable quickly
- Business constraints are needed to tell the story
  - Done well they are believable constraints
  - People can rally behind them for much longer

Arbitrary

- Pomodoro Timers
- Made up deadlines
- “must be written in Java”

Business

- Product must be cheap
- Protect against unpaid traffic
- Must support self hosting

NO CONSTRAINTS = ANYTHING GOES

- initially they seem the same:
  - config loaded by evaluating a touring complete language (config.js)
  - declarative files (config.json)
- over time, you realize the damage that this causes:
  - how can my IDE load it?
  - how can I provide assistance?
  - do i now need to sandbox it?
  - how do I cache the dynamic nature of it?

THIS CAUSES REAL PAIN

- pyproject.toml / setup.py in Python allow custom code execution
- compared to package.json which does not
- people started getting addicted to the dynamic nature and fight the removal
- a lot of complexity in resolvers, complex tooling necessary

- once it's there, how can you ever remove it?

INNER PLATFORM EFFECT & SECOND-SYSTEM SYNDROME

FIGHTING COMPLEXITY WITH MORE COMPLEXITY

- It starts with kubernetes
- It then incorporates helm
- Then someone adds custom tooling on top of it
- Then someone else adds tooling on top of the tooling

EACH ONION LAYER WORSE THAN THE OUTER

"The inner-platform effect is the tendency of software architects to create a system so customizable as to become a replica, and often a poor replica, of the software development platform they are using."

DECLARING TECH BANKRUPTCY

- Eventually people give up
- They ask for a fresh start
- They dismiss everything that has been created
- Including all the useful things in the mess
- And run at risk to not set the right constraints to prevent it from happening again

FIGHT IT EARLY
KEEP FIGHTING IT
IT'S WORTH IT